

DST-R8C

Delta System Tools for Renesas R8C Tiny

DST-R8C

BUKU PANDUAN PRAKTIKUM



Delta Electronic

<http://www.delta-electronic.com>

PENDAHULUAN

1. Tentang hardware

DST-R8C adalah salah satu development system yang dikeluarkan Delta Electronic. Development sistem ini dibuat dengan berbasis mikrokontroler R8C/Tiny Series yang diproduksi oleh Renesas Technology, Inc. Renesas sendiri adalah produsen semikonduktor yang merupakan gabungan dari Mitsubishi dan Hitachi.

Mikrokontroler R8C/Tiny Series termasuk keluarga M16C, dan mempunyai beberapa seri antara lain R8C/10, R8C/11, R8C/12, dll. Masing-masing seri berbeda dalam besarnya flash ROM dan jumlah pin I/O. Lebih terperinci mengenai spesifikasi masing-masing seri bisa dilihat pada general MCU presentation.pdf.

Mikrokontroler yang digunakan pada development system DST-R8C adalah R5F21134 yang termasuk seri R8C/13 dengan flash ROM sebesar 16 KB dan RAM internal sebesar 1 KB. Walaupun kecil ukurannya, Mikrokontroler ini mempunyai banyak fitur yang menarik, antara lain:

- Flash ROM dan Internal RAM relatif cukup besar
- Sudah ada ON Chip debugger
- Built-in ADC 10 bit 12 channel.
- 2 buah UART, 1 bisa dipakai pada mode sinkron
- 3 Timer 8 bit dan 1 Timer 16 bit (dengan prescaler dan beberapa mode kerja)
- Built-in internal clock
- dll

Sedangkan modul DST-R8C memiliki fasilitas antara lain:

- Tersedia 2 konektor untuk LCD, yakni untuk LCD Hitachi dan LCD standar.
- Tersedia konektor keyboard PS2
- Bisa mode Program dan Run (tinggal melepas atau memasang jumper)
- Bisa digunakan osilator clock internal atau eksternal (tinggal melepas atau memasang jumper). Ketika digunakan osilator internal, maka kaki Xin dan Xout bisa dipakai sebagai port I/O.
- Konektor I/O kompatibel dengan modul keypad 4x4.
- 4 buah saklar push button yang terhubung ke kaki Key Interrupt (KI0..KI3)
- RTC serial DS1307 lengkap dengan battery back-up
- EEPROM serial 1 Kb
- Port komunikasi RS485

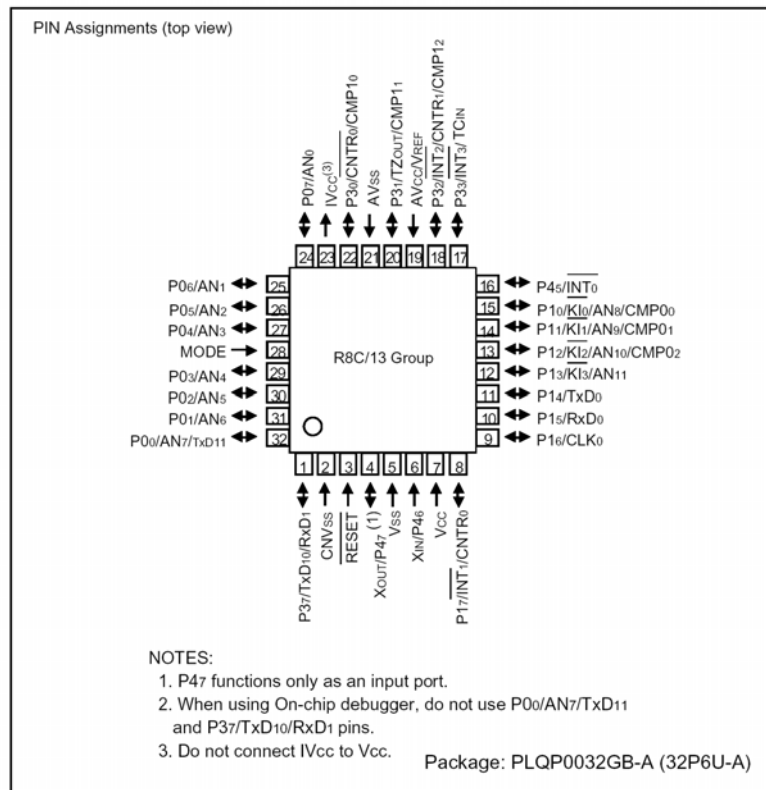
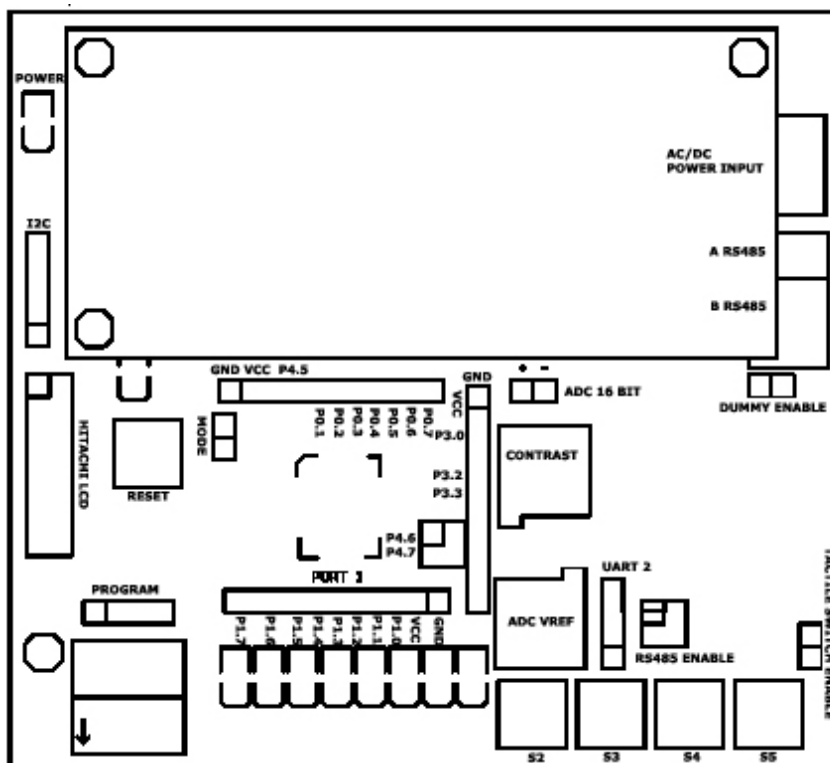


Figure 1.3 Pin Assignments (Top View)

Konfigurasi pin R5F21134

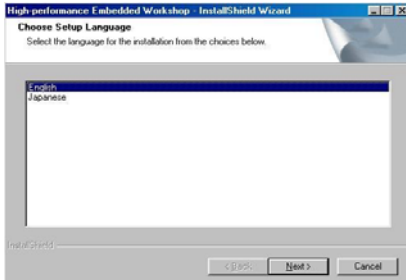


Tata letak komponen DST-R8C V3.0

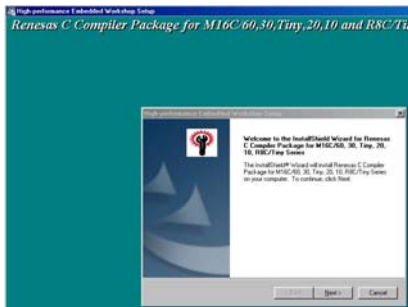
2. Tentang Software

a. C Compiler HEW

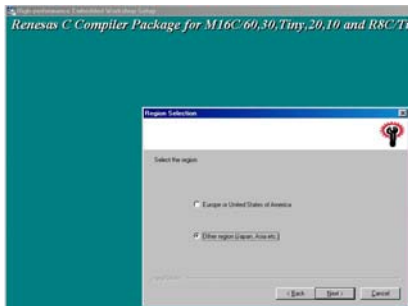
Berupa software IDE untuk membuat dan mengkompilasi program dalam bahasa C. Untuk menginstall program ini, jalankanlah file: **nc30wav530r02_2_ev.exe** yang ada di cd DST-R8C folder software.



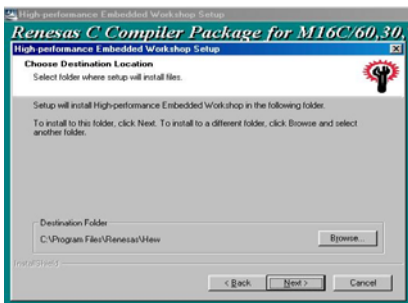
Pilih bahasa yang digunakan lalu Klik Next



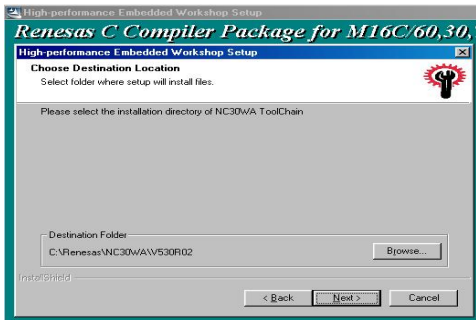
Klik Next



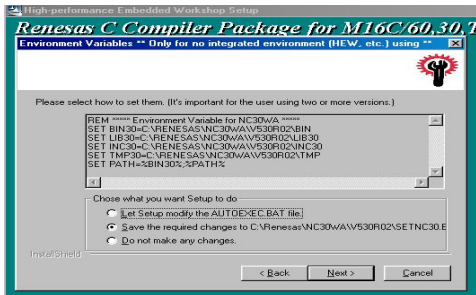
Pilih other region lalu klik next



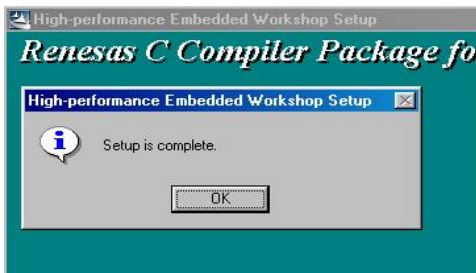
Pilih folder atau biarkan (recommended) lalu klik next



Pilih folder atau biarkan (recommended) lalu klik next



Pilih option atau biarkan (recommended) lalu Klik next



Instalasi selesai, klik OK

b. Debugger KD30

Berupa software untuk memasukkan program C yang telah dikompilasi ke flash ROM mikrokontroler dan untuk melakukan proses debug program yang telah diisikan. Software ini akan berkomunikasi dengan On-chip debugger yang sudah ada dalam mikrokontroler. Untuk menginstall program ini jalankanlah file: **KD30V410R1_E_20041203.exe** yang ada di cd DST-R8C folder software.



Klik OK



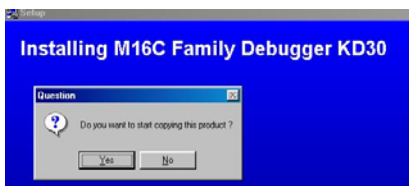
Klik next



Baca license agreement, jika setuju klik next



Pilih KD30 Debugger lalu klik next



Klik OK, tunggu sampai instalasi selesai

Setelah menginstall 2 program tersebut, dan dengan satu modul DST-R8C, kita telah siap untuk membuat project dengan mikrokontroler R8C/Tiny series.

Membuat Project

Langkah-langkah yang perlu dilakukan:

1. Instalasi modul
2. Membuat project barup
3. Menuliskan program
4. Mengisikan program ke flash ROM
5. Menjalankan / men-*debug* program yang telah diisikan.

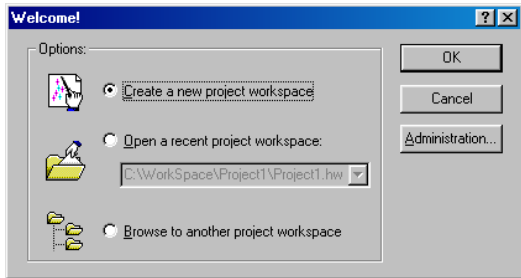
I. Instalasi modul

- Siapkan modul DST-R8C, kabel serial, power supply DC 12-14V
- Pasanglah kabel serial pada konektor P2, yaitu konektor putih 4 pin yang berada di belakang konektor PS2 dan pasang satu sisi lainnya pada port COM PC.
- Hubungkan konektor kabel power supply DC 12-14V pada JP10. Jika power telah masuk, maka LED indikator power (D1) akan menyala.
- Tekanlah tombol RESET (SW5). Tombol RESET harus selalu ditekan lebih dulu sebelum memasukkan. program

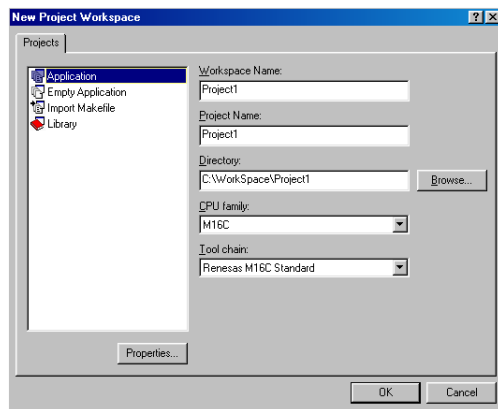
II. Membuat Project baru

Langkah-langkah yang dilakukan:

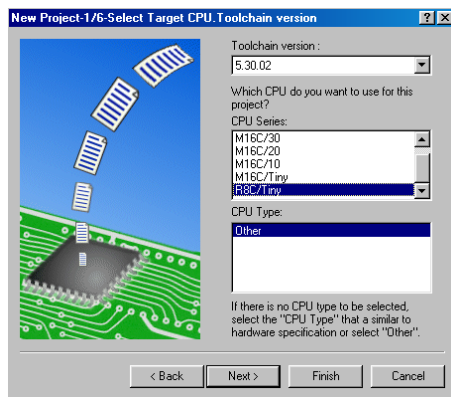
- Dari Start Menu klik: Start\Programs\Renesas\High Performance Embedded workshop\High Performance Embedded Workshop, maka akan muncul Form Welcome dengan options 3 pilihan.



- Pilih “Create a new project workspace”, lalu klik OK.

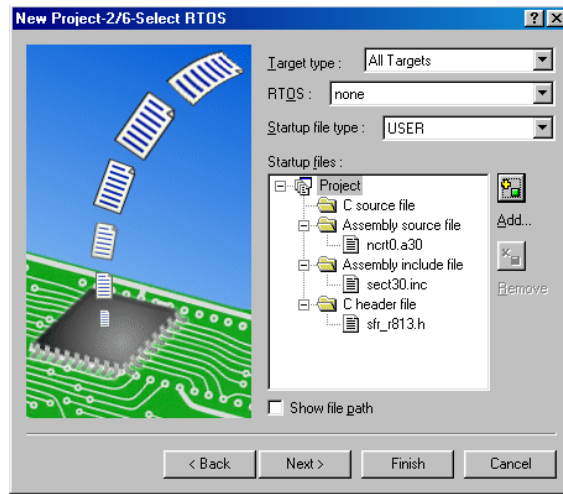


- Isikan nama Project pada Workspace Name (misal: Project1), maka yang lain akan mengikuti, Setelah itu Klik OK.

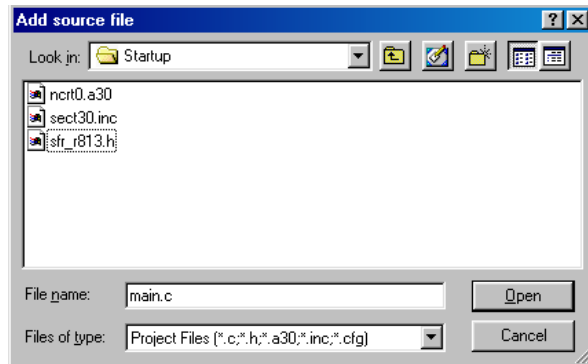


- Papada CPU series pilih R8C/Tiny, lalu klik Next

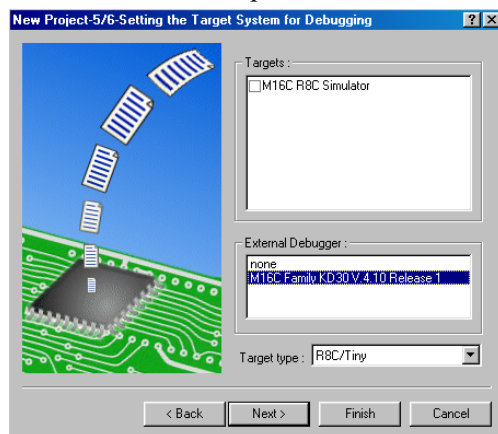
- Pilih Startup file type dengan “USER”. Startup file secara otomatis akan di-include-kan ke project. Klik tombol “ADD” dan masuklah kedalam folder Startup.



- Pada folder ini sudah kami kumpulkan beberapa file yang diperlukan untuk startup. Bukalah ketiga file ini, dan file yang akan di sertakan di dalam project.

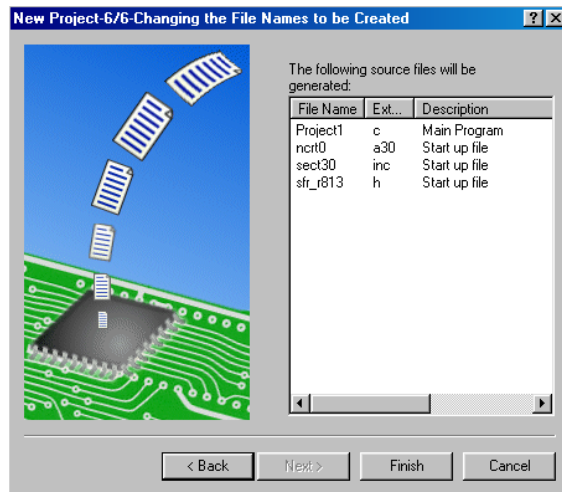


- Klik NEXT sampai muncul form berikut



- Pada Form berikutnya ubahlah parameter External Debugger: ”M16C Family KD30V.4.10Release 1”, lalu klik “Next”.

- Form berikutnya klik finish, maka akan muncul form summary. Klik OK, maka lembar kerja program terbuka, dan kita siap menulis program



III. Menulis Program

Program yang kita tulis adalah dalam bahasa C. Berikut akan diberikan contoh program sederhana running led. 8 led akan dipasang pada port1. Ketikkanlah program dibawah ini:

```
#include "sfr_r813.h"
#include "atur_clk.h"

void init();
void Delay(long Time);

void main(void)
{
    unsigned char temp = 0x01;
    init();
    while (1)
    {
        p1 = ~temp;
        temp = temp << 1;
        if (temp == 0x00) temp = 0x01;
        Delay(500000);
    }
}

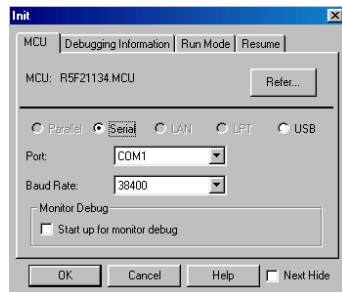
void Delay(long Time)
{while (Time--);}
```

IV. Mengisikan program ke flash ROM dan menjalankannya

Sebelum program yang telah kita ketik kita masukkan ke flash ROM, sebelumnya harus dilakukan proses build dengan menekan **F7** atau **Build → Build**. Dengan proses build ini sekaligus bisa diketahui ada tidaknya kesalahan program. Jika program yang telah kita tuliskan benar, maka akan ada pesan: 0 Errors, 0 Warning.

Setelah program yang kita tuliskan benar, maka hasil kompilasi siap dimasukkan ke dalam flash ROM. Langkah-langkahnya:

- Pilih **Tools → Launch External Debugger** maka akan muncul:



Pilihlah Port COM PC yang anda gunakan untuk memprogram flash ROM. Untuk baud rate biarkan ada default tertinggi 38400 bps. Tombol Refer untuk memilih file firmware untuk komunikasi MCU – PC, karena itu, pilihlah yang sesuai dengan MCU yang digunakan, dalam hal ini R5F21134. File ini terdapat pada folder: McuFiles\R8C-Tiny Series\R5F21134.MCU.

Setelah pengaturan-pengaturan tersebut, kliklah tombol OK. Perlu diperhatikan lagi, sebelum klik Tombol OK tekanlah switch RESET terlebih dahulu. Setelah itu akan terjadi proses downloading firmware dan program. Untuk menjalankan program (debug) pilihlah **Go**. Untuk tracing program perbaris, pilihlah **Step** atau **Over**.

Jika pembuatan program telah selesai, lepas jumper mode dan tekan reset untuk menjalankan..

1. Clock dan I/O (input/ output)

Sumber dari clock utama adalah rangkaian resonator (crystal dan kapasitor-kapasitor) yang dipasang pada kaki P46 dan P47 (Xin-Xout). Nilai crystal bisa kita pilih sampai pada frekuensi 20MHz. Sedangkan Osilator internal sumbernya berasal dari rangkaian internal dan tidak memerlukan komponen tambahan.

	Main Clock Oscilator	On Chip Oscilator	
		Hig Speed	Low speed
Fungsi	-Sumber Clock CPU -Sumber Clock Peripheral	- Sumber Clock CPU - Sumber Clock CPU dan Peripheral ketika main clock mati	
Frekuensi	0 – 20 MHz	± 8 MHz	± 125 KHz
Oscilator yang digunakan	- Ceramik - Crystal	---	---
Pin yang digunakan	Xin dan X out	---	---
Setelah reset	- Mati	- Mati	-Hidup

CM0 (System clock control register 0)			
Bit ke-	Simbol	Nama bit	Fungsi
0	----	Tidak dipakai	
1	----		
2	cm02	WAIT peripheral function clock stop bit	0 : Peripheral tidak berhenti ketikan mode wait 1 : Peripheral berhenti ketikan mode wait
3	----	Tidak dipakai	
4	----		
5	cm05	Main Clock	0 : on 1 : off
6	cm06	Pembagi Clock	0 : CM16 and CM17 valid 1 : dibagi 8
7	----	Tidak dipakai	

CM0 (System clock control register 1)			
Bit ke-	Simbol	Nama bit	Fungsi
0	cm10	Stop bit untuk semua clock	0 : Clock ON 1 : Semual clock mati (stop mode)
1	----	Tidak dipakai	
2	----		
3	cm13	Pemilih Xin - Xout	0 : P46 dan P47 sebagai input 1 : P46 dan P47 sebagai Xin - Xout
4	cm14	stop bit untuk low speed on chip oscilator	0 : low speed oscilator on 1 : low speed oscilator off
5	cm15	Xin – Xout drive capacity	0 : Low 1 : High
6	cm16	Pembagi Clock	0 0 : dibagi 1 (tidak dibagi) 0 1 : dibagi 2
7	cm17		1 0 : dibagi 4 1 1 : dibagi 16

OCD (Oscilation stop detection register)			
Bit ke-	Simbol	Nama bit	Fungsi
0	ocd0	Oscilation stop detection enable bit	0 0 : disable 0 1 : Do not set
1	ocd1		1 0 : Do not set 1 1 : Enable
2	ocd2	Pemilih System clock	0 : Main clock 1 : On chip oscilator clock
3	ocd3	Monitor clock	0 : Main clock on (read only) 1 : Main clock off (read only)
4	----	Tidak di pakai	
5	----		
6	----		
7	----		

HR0 (High speed on chip oscilator control register 0)			
Bit ke-	Simbol	Nama bit	Fungsi
0	hr00	High speed on chip oscilator enable bit	0 : off 1 : on
1	hr01	Bit pemilih high speed on chip oscilator	0 : Low speed 1 : High speed
2	----	Tidak di pakai Diisi dengan " 0"	
3	----		
4	----		
5	----		
6	----		
7	----		

HR1 (High speed on chip oscilator control register 1)			
Pengatur frekuensi high speed on chip oscilator			
$\text{Period} = \text{td}(\text{HR offset}) + (64\text{x}b6 + 32\text{x}b5 + 16\text{x}b4 + 8\text{x}b3 + 4\text{x}b2 + 2\text{x}b1 + b0) \times \text{td}(\text{HR})$			
b0 to b6 : Bits in HR1 register			

Clock utama mulai Osilasi ketika bit **cm05** (main clock on/off) yang ada didalam register **cm0** diberi nilai "0" (clock utama on) serta bit **cm13** (Pemilih Xin - Xout) dan bit **cm14** (stop bit untuk low speed on chip oscilator) yang ada didalam register **cm1** diberi nilai "1".

Untuk menggunakan clock utama sebagai CPU clock, isilah bit **ocd2** yang ada di dalam register **ocd** dengan "0" setelah clock utama stabil.

CPU Clock dan Peripheral Function Clock

CPU clock digunakan untuk CPU dan Watchdog timer, clock yang di pilih (on chip atau main) frekuensinya dapat diatur, yakni dengan di bagi dengan 1, 2, 4, 8 atau 32. Misalnya crystal yang di gunakan adalah 20 MHz, sedangkan clock yang diinginkan adalah 10 MHz, maka yang harus di lakukan adalah membaginya dengan 2 yakni 20 MHz dibagi 2.

Setelah reset mikrokontroler akan menggunakan low speed on chip oscilator dibagi 8 sebagai CPU Clock dan Peripheral clock.

- f_i ($i = 1, 2, 4, 8$ dan 32) di bangkitkan oleh main clock atau on chip oscilator, clock ini dapat digunakan untuk CPU clock dan count source timer X, Y, Z dan C.
- f_{jSIO} ($j = 1, 8$ dan 32) di bangkitkan oleh main clock atau on chip oscilator, clock ini digunakan untuk count source baudrate serial.
- f_{AD} di bangkitkan oleh main clock atau on chip oscilator, clock ini digunakan untuk Analog to digital Converter.
- f_{RING} dan $f_{RING 128}$, frekuensi f_{RING} sama dengan frekuensi on chip oscilator clock, clock ini dapat digunakan untuk count source timer Y, sedangkan $f_{RING 128}$ berasal dari f_{RING} dibagi 128, $f_{RING 128}$ dapat di gunakan untuk count source timer C. Dalam mode WAIT f_{RING} dan $f_{RING 128}$ tetap hidup.
- $f_{RING-fast}$ berasal dari high speed onchip oscilator dan **hr00** di beri nilai "1" (high speed on chip oscilator on) . Dalam mode WAIT $f_{RING - fast}$ tetap hidup.

Mode		OCD	CM1		CM0	
		ocd2	cm17	cm16	cm06	cm05
High speed		0	0	0	0	0
Medium speed	dibagi 2	0	0	1	0	0
	dibagi 4	0	1	0	0	0
	dibagi 8	0	----	----	1	0
	dibagi 16	0	1	1	0	0
High speed, low speed on chip oscilator	dibagi 1	1	0	0	0	x
	dibagi 2	1	0	1	0	x
	dibagi 4	1	1	0	0	x
	dibagi 8	1			1	x
	dibagi 16	1	1	1	0	x

* Low speed oscilator **cm14 = 0** dan **hr01 = 0**

* High speed oscilator **hr00 = 2** dan **hr01 = 1**.

Protect register

Protect register berguna untuk melindungi beberapa register penting dari penulisan yang tidak di ijinakan, beberapa register yang mempunyai proteksi diantaranya :

- PRC0 bit: CM0, CM1, and OCD, HR0, HR1 registers
- PRC1 bit: PM0 and PM1 registers
- PRC2 bit: PD0 register
- PRC3 bit: VCR2 and D4INT registers

PRCR (Protect register)			
Bit ke-	Simbol	Nama bit	Fungsi
0	prcr0	Protect bit 0	0 : Write protect 1 : Write enable
1	prcr1	Protect bit 1	
2	prcr2	Protect bit 2	
3	prcr3	Protect bit 3	

I/O (input/ output)

Pada Development System DST-R8C ada 4 port yang bisa kita gunakan untuk berhubungan dengan divais luar, port-port tersebut adalah sebagai berikut :

Port 0 8 bit (P00..P07) - Input/Output

Port 1 8 bit (P10..P17) - Input/Output

Port 3 5 bit (P30..P33, P37) - Input/Output

Port 4 1 bit (P45) – Input/Output; 2 bit (P46, P47) – Hanya Input

Sebelum digunakan, port-port tersebut harus didefinisikan dahulu, apakah akan difungsikan sebagai input, atau akan difungsikan sebagai output. Dengan cara mengisi register port direction (pd) dengan **0** apabila sebagai input dan **1** apabila sebagai output.

PDi (Port direction register i = 0,1,3,4)			
Bit ke-	Simbol	Nama bit	Fungsi
0	pdi_0	port i direction bit 0	0 = input 1 = output
1	pdi_1	port i direction bit 1	
2	pdi_2	port i direction bit 2	
3	pdi_3	port i direction bit 3	
4	pdi_4	port i direction bit 4	
5	pdi_5	port i direction bit 5	
6	pdi_6	port i direction bit 6	
7	pdi_7	port i direction bit 7	

PUR0 (Pull up control register 0)			
Bit ke-	Simbol	Nama bit	Fungsi
0	pu00	p00 – p03 pull up	1 = pull up
1	pu01	p04 – p07 pull up	
2	pu02	p10 – p13 pull up	
3	pu03	p14 – p17 pull up	
4	----		
5	----		
6	pu06	p30 – p33 pull up	
7	pu07	p37 pull up	

PUR1 (Pull up control register 1)			
Bit ke-	Simbol	Nama bit	Fungsi
0	----		1 = pull up
1	pu11	p45 pull up	

** keterangan lengkap lihat hard ware manual r813.pdf bagian programable I/O port.

Praktikum 1 I/O dan setting clock

Tujuan

- Mempelajari cara mengatur port
- Mempelajari cara penggunaan clock eksternal dan pembagiannya

Peralatan

- Modul DST-R8C v.3

Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Bukalah project praktek_io_clk
File => open workspace => praktek_io_clk.hws
3. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
4. Masukkan program ke flash Rom dan jalankan.
5. Perhatikan tampilan LED
6. Rubah nilai dari **pd1** (baris 39) dengan 0xf0, atau sesuai keinginan (00 sampai FF)
7. Ulangi langkah 3,4 dan 5.
8. Perhatikan kecepatan nyala dan padam LED
9. Rubah nilai dari **CM06, CM16, CM17** (baris 49).
10. Ulangi langkah 3,4 dan 5.
11. Perhatikan kecepatan nyala dan padam LED

2. LCD 16 X 2 BARIS

Dalam modul DST-R8C terdapat 2 buah konektor yang di desain khusus untuk menghubungkan LCD dan mikrokontroler.

M1632 adalah merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya yang rendah. Modul ini dilengkapi dengan mikrokontroler yang didisain khusus untuk mengendalikan LCD. Mikrokontroler HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD ini mempunyai CGROM (Character Generator Read Only Memory), CGRAM (Character Generator Random Access Memory) dan DDRAM (Display Data Random Access Memory).

DDRAM adalah merupakan memori tempat karakter yang ditampilkan berada. Contoh, untuk karakter 'A' atau 41H yang ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karakter tersebut ditulis di alamat 40, maka karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD.

Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
	HD44780U display								Extension driver display							
For shift left	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50
For shift right	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E

DDRAM M1632 (diambil dari data sheet HD44780)

CGRAM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana bentuk dari karakter dapat diubah-ubah sesuai keinginan. Namun memori ini akan hilang saat power supply tidak aktif, sehingga pola karakter akan hilang.

CGROM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana pola tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubah lagi. Namun karena ROM bersifat permanen, maka pola karakter tersebut tidak akan hilang walaupun power supply tidak aktif

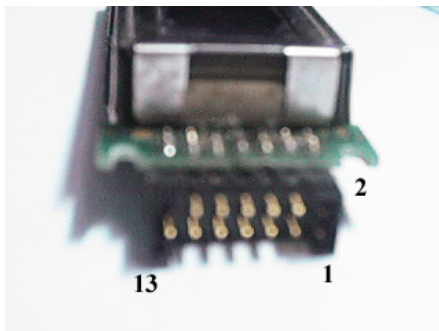
Pada gambar 2, tampak terlihat pola-pola karakter yang tersimpan dalam lokasi-lokasi tertentu dalam CGROM. Pada saat HD44780 akan menampilkan karakter 'A' atau 41H , maka HD44780 akan mengambil data di alamat 41H (0100 0001) yang ada pada CGROM yaitu pola karakter A.

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111		
xxxx0000	(1)		0	@P`P									-	ヲ	ミ	αp		
xxxx0001	(2)		!	1AQa9									。	ア	チ	ㄥ	äq	
xxxx0010	(3)		"	2BRbr									「	イ	ツ	×	βθ	
xxxx0011	(4)		#	3CScs									」	ウ	テ	モ	εω	
xxxx0100	(5)		\$	4DTdt									、	エ	ト	ト	μΩ	
xxxx0101	(6)		%	5EUeu									。	オ	ナ	1	εÜ	
xxxx0110	(7)		&	6FUfu									ヲ	カ	ニ	ヨ	ρΣ	
xxxx0111	(8)		'	7GWgw									フ	キ	ヌ	ウ	gπ	
xxxx1000	(1)		<	8HXhx									イ	ク	ネ	リ	フ	α
xxxx1001	(2)		>	9IYiy									ウ	ケ	ノ	ル	レ	γ
xxxx1010	(3)		*	:JZJz									エ	コ	ン	レ	j	κ
xxxx1011	(4)		+	;K[k<									オ	サ	ヒ	ロ	*	κ
xxxx1100	(5)		,	<L¥ll									カ	シ	フ	ワ	φ	κ
xxxx1101	(6)		-	=M]m>									ユ	ズ	ヘ	ン	モ	÷
xxxx1110	(7)		.	>N^n÷									ヨ	セ	ホ	°	ñ	
xxxx1111	(8)		/	?0_0€									ウ	ツ	マ	°	ö	■

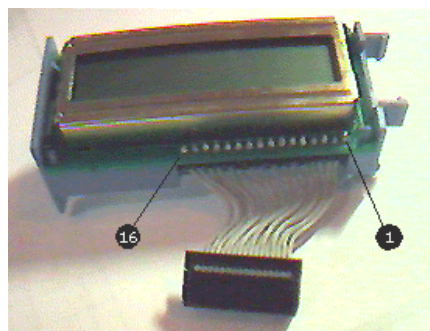
Isi dari CGROM

Pin Out LCD

No	Nama Pin	Deskripsi
1	VCC	+5V
2	GND	0V
3	VEE	Tegangan Kontras LCD
4	RS	Register Select, 0 = Register Perintah, 1 = Register Data
5	R/W	1 = Read, 0 = Write
6	E	Enable Clock LCD, logika 1 setiap kali pengiriman atau pembacaan data
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7
15	Anoda (Kabel coklat untuk LCD Hitachi)	Tegangan positif backlight
16	Katoda (Kabel merah untuk LCD Hitachi)	Tegangan negatif backlight



Pin Out M1632 LCD Hitachi



Pin Out LCD M1632 Standard

Register

M1632, mempunyai dua buah Register yang aksesnya diatur dengan menggunakan kaki RS. Pada saat RS berlogika 0, maka register yang diakses adalah Register Perintah dan pada saat RS berlogika 1, maka register yang diakses adalah Register Data

Register Perintah

Register ini adalah register di mana perintah-perintah dari mikrokontroler ke M1632 pada saat proses penulisan data atau tempat status dari M1632

Perintah-perintah M1632

Perintah	D7	D6	D5	D4	D3	D2	D1	D0	Deskripsi
Hapus Display	0	0	0	0	0	0	0	1	Hapus Display dan DDRAM
Posisi Awal	0	0	0	0	0	0	1	X	Set Alamat DDRAM di 0
Set Mode	0	0	0	0	0	1	I/D	S	Atur arah pergeseran cursor dan display
Display On/OFF	0	0	0	0	1	D	C	B	Atur display (D) On/OFF, cursor (C) ON/OFF, Blinking (B)
Geser Cursor/Display	0	0	0	1	S/C	R/L	X	X	Geser Cursor atau display tanpa merubah alamat DDRAM
Set Fungsi	0	0	1	DL	N	F	X	X	Atur panjang data, jumlah baris yang tampil, dan font karakter
Set Alamat CGRAM	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Data dapat dibaca atau ditulis setelah alamat diatur
Set Alamat DDRAM	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Data dapat dibaca atau ditulis setelah alamat diatur

X = diabaikan

I/D 1=Increment, 0=Decrement

S 0=Display tidak geser

S/C 1=Display Shift, 0=Geser Cursor

R/L 1=Geser Kiri, 0=Geser Kanan

DL 1=8 bit, 0=4bit

N 1=2 baris, 0=1 baris

F 1=5x10, 0=5x8

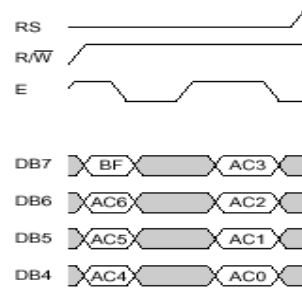
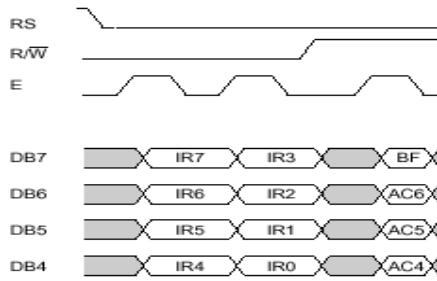
D 0=Display OFF, 1=Display ON

C 0=Cursor OFF, 1=Cursor ON

B 0=Blinking OFF, 1=Blinking ON

Delta Electronic

<http://www.delta-electronic.com>

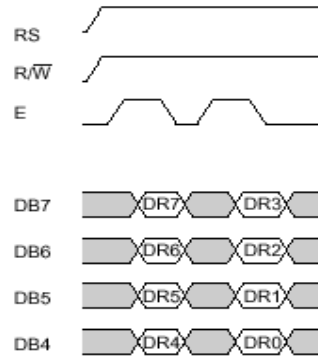
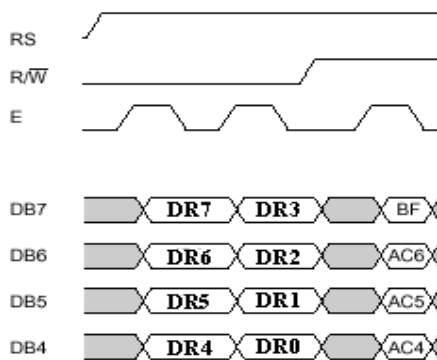


Penulisan Data ke Register Perintah

Pembacaan Data dari Register Perintah

Register Data

Register ini adalah register di mana mikrokontroler dapat menuliskan atau membaca data ke atau dari DDRAM. Penulisan data pada register ini akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya



Penulisan Data ke Register Data

Pembacaan Data dari Register Data

Praktikum 2 LCD M1632

Tujuan

- Menampilkan karakter pada LCD

Peralatan

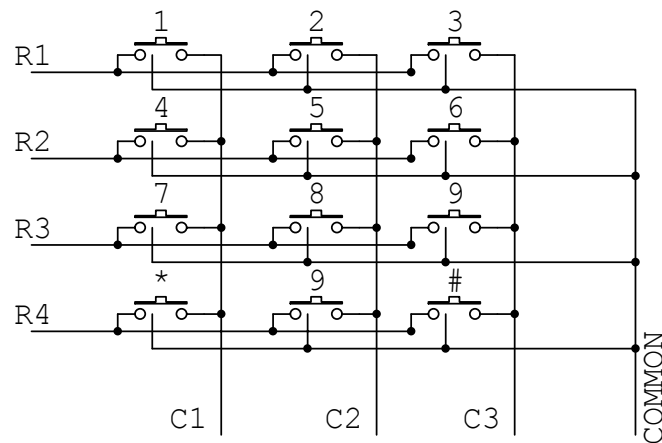
- Modul DST-R8C v.3
- LCD M1632

Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Pasang LCD ke modul DST-R8C sesuai dengan jenis LCD yang digunakan
3. Bukalah project praktek_lcd
File => open workspace => praktek_lcd.hws
4. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
5. Masukkan program ke flash Rom dan jalankan.
6. Perhatikan tampilan LCD
7. Gantilah **Kirim_Karakter('A');** (baris 29) dengan **Kirim_Karakter(0x41);**
8. Ulangi langkah 3,4 dan 5.

3. Keypad 4 x 3(common)

Jika digambarkan dengan diagram skematik, maka kita akan mendapatkan gambar konstruksi hardwarenya seperti berikut



Konstruksi Keypad 4x3 dengan common

Jika pin common di hubungkan ke logika 0 maka setiap tombol ditekan akan memberikan logika 0 pada baris dan kolom tertentu.

Tombol	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1
1	0	1	1	0	1	1	1	0	7C
2	1	0	1	0	1	1	1	0	7A
3	1	1	0	0	1	1	1	0	76
4	0	1	1	1	0	1	1	0	BC
5	1	0	1	1	0	1	1	0	BA
6	1	1	0	1	0	1	1	0	B6
7	0	1	1	1	1	0	1	0	DC
8	1	0	1	1	1	0	1	0	DA
9	1	1	0	1	1	0	1	0	D6
0	0	1	1	1	1	1	0	0	EC
*	1	0	1	1	1	1	0	0	EA
#	1	1	0	1	1	1	0	0	E6

⇒ port **p1_0** sebagai common

Praktikum 3

KEYPAD 4 x 3 (common)

Tujuan

- Menampilkan karakter pada LCD sesuai dengan tombol keypad yang di tekan

Peralatan

- Modul DST-R8C v.3
- LCD M1632
- Modul Keypad 4 x 3 dengan common

Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Pasang LCD ke modul DST-R8C sesuai dengan jenis LCD yang digunakan
3. Pasang modul keypad 4 x 3 di port 1, common (garis putih) di hubungkan ke port 1.0
4. Bukalah project praktek_keypad
File => open workspace => praktek_keypad.hws
5. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
6. Masukkan program ke flash Rom dan jalankan.
7. Tekan tombol yang ada didalam modul keypad
8. Perhatikan tampilan LCD dan LED

Praktikum 4 PC KEYBOARD

Tujuan

- Mengambil data dari PC Keyboard
- Mengenali scan code dan ASCII code

Peralatan

- Modul DST-R8C v.3
- Modul LCD M1632
- PC Keyboard

Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Hubungkan keyboard melalui konektor PS2
9. Pasang LCD ke modul DST-R8C sesuai dengan jenis LCD yang digunakan
3. Bukalah project praktek_keyboard
File => open workspace => praktek_keyboard.hws
4. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
5. Masukkan program ke flash Rom dan jalankan.
6. Tekan salah satu tombol keyboard (huruf atau angka saja)
7. Amati tampilan LCD

5. Konversi dari Analog ke Digital

Modul DST-R8C mempunyai 12 kanal ADC yang bisa dihubungkan langsung dengan tegangan analog 0 sampai 5 Volt. Register-register yang harus diatur dalam penggunaan ADC adalah **ADCON0**, **ADCON1** dan **ADCON2**. Keterangan tentang pengaturan register-register ini bisa dilihat pada tabel 5

Hasil konversi ADC ditampung pada register **AD**. ADC mulai bekerja ketika **ADST** diberi nilai 1.

Tabel 5.a

Adcon0			0	1
bit 0	Ch0	Pemilih Analog pin	Lihat tabel 5.b Pemilih channel	
bit 1	Ch1			
bit 2	Ch2			
bit 3	md	Mode operasi	One shot	repeat
bit 4	adgsel0	Pemilih grup input	AN0 samapai AN7	AN8 sampai AN11
bit 5	Tidak di pakai	----	Harus 0	
bit 6	adst	Start dan stop	ADC tidak aktif	ADC mulai konversi
bit 7	cks0	Pemilih frekuensi	fAD /4	fAD /2

Tabel 5.b

ch2	ch1	ch0	adgsel0 = 0	adgsel0 = 1
0	0	0	AN0	
0	0	1	AN1	
0	1	0	AN2	
0	1	1	AN3	
1	0	0	AN4	AN8
1	0	1	AN5	AN9
1	1	0	AN6	AN10
1	1	1	AN7	AN11

Tabel 5.c

Adcon1			0	1
bit ke	simbol	fungsi	harus di isi 0	
bit 0	----	tidak di pakai		
bit 1	----			
bit 2	bits	8 atau 10 bit	8 bit	10 bit
bit 3	cks1	pemilih frekuensi	cks0 valid	fad di pilih
bit 4	vcut	vref	vref tidak terhubung	vref terhubung
bit 5	----	tidak di pakai	harus di isi 0	
bit 6	----			
bit 7	----			

Tabel 5.d

Adcon1			0	1
bit	simbol	fungsi	Tanpa sample dan hold	sample dan hold
bit 1	smp	Metode konversi		

* hanya bit 1 yang di pakai, yang lain disisi dengan 0

Praktikum 5 Konversi dari Analog ke Digital

Tujuan

- Mengambil data analog (berupa tegangan 0 sampai 5 volt)
- Menampilkan hasil konversi ke display (LCD)
- Membuat Volt meter digital

Peralatan

- Modul DST-R8C v.3
- Modul LCD M1632
- Photensio ($\pm 10\text{ K}$) dan kabel secukupnya

Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Hubungkan photensio dan DST-R8C sesuai tabel berikut

Photensio	DST-R8C
Pin 1	Ground
Pin 2	P0_1
Pin 3	Vcc (5 volt)

3. Pasang LCD ke modul DST-R8C sesuai dengan jenis LCD yang digunakan
4. Bukalah project praktek_adc
File => open workspace => praktek_adc.hws
5. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
6. Masukkan program ke flash Rom dan jalankan.
7. Putar knob Photensio
8. Ukur tegangan yang masuk ke pin P0_1
9. Bandingkan hasil ukur dengan yang tertera di LCD

6. TIMER X

MCU R8C/Tiny Series mempunyai 3 buah timer 8 bit (Timer X, Timer Y, Timer Z) yang masing-masing mempunyai prescaler 8-bit, dan satu buah Timer 16 bit (Timer C) .

Timer X adalah timer 8-bit yang juga mempunyai prescaler 8-bit. Timer ini memiliki lima mode kerja sebagai berikut:

- **Timer** : Timer menghitung dengan referensi count source internal
- **Pulse Output** : Timer menghitung dengan referensi count source internal, dan ketika terjadi underflow maka akan dikeluarkan sebuah pulsa pada pin tertentu (P30)
- **Event Counter** : Timer menghitung pulsa yang berasal dari sumber external(P30)
- **Pulse Width Measurement** : Timer menghitung lebar suatu pulsa external(P30)
- **Pulse Periode Measurements** : Timer menghitung periode suatu pulsa external(P30)

TXMR (Timer X Mode Register)

Bit ke-	Simbol	Nama bit	Fungsi
0	txmod0	Pemilih mode operasi Bit 0,1	0 0 : Mode Timer / pulse periode measurement 0 1 : Mode pulse output
1	txmod1		1 0 : Mode Event counter 1 1 : Mode Pulse width measurement
2	r0edg	INT1 / CNTR0 Pemilih polaritas	0 : Rising edge 1 : Fallin edge
3	txs	Timer x start/stop flag	0 : stop 1 : start
4	tx0cnt	P30/CNTR0	Fungsi tergantung mode operasi
5	txmod2	Pemilih mode operasi Bit 2	0 : Kecuali Pulse periode measurement 1 : Pulse periode measurement
6	txedg	Active edge reception flag	Fungsi tergantung mode operasi
7	txund	Timer x under flow flag	Fungsi tergantung mode operasi

TCSS (Timer Count Source Setting)

Bit ke-	Simbol	Nama bit	Fungsi
0	txck0	Bit pemilih source Timer x	0 0 : f1 0 1 : f8
1	txck1		1 0 : f32 1 1 : f2
2	tyck0	Bit pemilih source Timer y	0 0 : f1 0 1 : f8
3	tyck1		1 0 : fRING 1 1 : CNTR1
4	tzck0	Bit pemilih source Timer z	0 0 : f1 0 1 : f8
5	tzck1		1 0 : Timer Y underflow 1 1 : f2
6	-----	Harus di set 0	
7	-----		

Contoh inisialisasi ketika menggunakan Timer X pada mode Timer adalah:

- TXMR diisi nilai 0x00
- PREX dan TX diisi nilai preset bagi Timer X, dengan menggunakan rumus:

$$T = \text{count source} * (\text{Prescaler X} + 1) * (\text{Timer X} + 1)$$

Dimana:

- T : Periode yang diinginkan
- Prescaler : Nilai yang diisikan ke register PREX
- Timer X : Nilai yang diisikan ke register TX

- TXCK0 dan TXCK1 diisi nilai yang menentukan pembagian sumber clock referensi bagi Timer X.

Praktikum 6

TIMER X Mode Timer

Tujuan

- Menggunakan Timer X untuk delay

Peralatan

- Modul DST-R8C v.3

Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Bukalah project praktek_timer
File => open workspace => praktek_timer.hws
3. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
4. Masukkan program ke flash Rom dan jalankan.
5. Rubah nilai prex dan tx (baris 27 dan 28), nilai harus antara 1 sampai 255
6. Ulangi langkah 3,4 dan 5.
7. Perhatikan kecepatan nyala dan padam LED(port 1.0)

7. Universal Asynchronous Transmitter/Receiver (UART)

MCU R8C/Tiny Series mempunyai 2 channel UART (UART0 dan UART1). Secara default UART1 digunakan untuk pengisian program (isp) dan debugger, jika UART1 digunakan maka fungsi debugger tidak dapat digunakan namun isp masih bisa digunakan. Register yang berhubungan dengan UART1 dan UART0 antara lain:

UiMR (UART i Mode register i = 0 dan 1)

Bit ke-	Simbol	Nama bit	Fungsi
0	smd0	serial interface mode	0 0 0 : disable
1	smd1		0 0 1 : sinkron
2	smd2		0 1 0 : UART, 7 bit 0 1 1 : UART, 8 bit 1 1 0 : UART, 9 bit
3	ckdir	internal / eksternal clock (untuk baud rate)	0 : internal clock 1 : eksternal clock
4	stps	panjang stop bit	0 : 1 bit 1 : 2 bit
5	pry	odd / even parity	0 : odd (jika pry = 1) 1 : event (jika pry = 1)
6	prye	parity enable	0 : disable 1 : enable
7	----	----	di set 0

UiC0 (UART i transmit/receive control register i = 0 dan 1)

Bit ke-	Simbol	Nama bit	Fungsi
0	clk0	BRG cout source	0 0 : f1 SIO
1	clk1		0 1 : f8 SIO 1 0 : f32 SIO
2	----	----	set 0
3	txept	Transmit register empty flsg	0 : sedang transimisi data 1 : transmisi selesi
4	----	----	set 0
5	nch	Data output	0 : txd sebagai CMOS 1 : txd sebagai N-channel open-drain
6	ckpol	CLK polarity	0 : fallin edge 1 : rising edge
7	unform	Transmit format	0 : LSB first 1 : MSB first

UiC1 (UART i transmit/receive control register i = 0 dan 1)			
Bit ke-	Simbol	Nama bit	Fungsi
0	te	Transmit enable	0 : disable 1 : enable
1	ti	Transmit buffer empty flag	1 : empty
2	re	Receive enable bit	0 : disable 1 : enable
3	ri	Receive complete flag	0 : complete
4	----	Tidak dipakai	
5	----		
6	----		
7	----		

UCON (UART transmit/receive control register 2)			
Bit ke-	Simbol	Nama bit	Fungsi
0	u0irs	UART 0 transmit interrupt cause bit	0 : Transmit buffer empty (ti = 1) 1 : Transmission complete (texpt = 1)
1	u1irs	UART 1 transmit interrupt cause bit	0 : Transmit buffer empty (ti = 1) 1 : Transmission complete (texpt = 1)
2	uorrm	UART continuous receive mode enable bit	0 : disable 1 : enable
3	----	set to 0	
4	----		
5	txdsel	Port TxD11 switching bit	0 : I/O port p00 1 : TxD11
6	txd1en	TxD10/xD1 select bit	0 : RxD1 1 : TxD10
7	---	Tidak dipakai	

Port yang digunakan untuk UART 0 dan UART 1:

Port	Fungsi	Keterangan
P1_4	TxD0 / data output	P1_4 di set sebagai output (Pd1_4 = 1)
P1_5	RxD0 / data input	P1_5 di set sebagai input (Pd1_5 = 0)
P3_7	RxD10 / TxD1	TXD1EN=1 TXD1EN=0, PD3 register PD3_7 bit=0
P0_0	TxD1	Serial data output, TXD1SEL=1

Baud Rate :

Baud rate dibangkitkan oleh baud rate generator dan dapat diatur dengan mengisi nilai tertentu pada register **u0brg**, dengan nilai seperti pada tabel berikut (x-tal 20 MHz).

Baud Rate (bps)	Count source	u0brg
1200	f8	129
2400	f8	64
4800	f8	32
9600	f1	129
14400	f1	86
19200	f1	64
28800	f1	42
31250	f1	39
38400	f1	32
51200	f1	23

Atau dapat dihitung dengan menggunakan rumus berikut :

Jika memakai internal clock

$$u0brg = (f_j / (\text{bit rate} \times 16)) - 1$$

f_j : count source (f1SIO, f8SIO dan f32SIO)

Jika memakai eksternal clock (main clock)

$$u0brg = (f_{EXT} / (\text{bit rate} \times 16)) - 1$$

f_{EXT} : Count source (external clock)

Praktikum 7
UART
Universal Asynchronous Transmitter/Receiver

Tujuan

- Komunikasi data antara PC dan Modul DST-R8C
- Menampilkan data yang di terima ke display (LCD)
- Setting baud rate

Peralatan

- 2 buah modul DST-R8C v.3
- 2 buah modul LCD M1632
- 3 utas kabel (dengan konektor)

Prosedure Praktikum

1. Instalasi kedua buah modul, pastikan modul pada mode program.
2. Pasang LCD ke modul DST-R8C sesuai dengan jenis LCD yang digunakan
3. Hubungkan UART 0 DST I dan UART 0 DST II seperti pada tabel berikut

DST-R8C I	DST-R8C II
TXD	RXD
RXD	TXD
GND	GND

4. Untuk modul DST-R8C I bukalah project praktek_serial_A
 - **File => open workspace => praktek_serial_A.hws**
 - Build **dengan** menekan **F7** atau masuk menu **build => build**
 - **Build All** jika pertama kali buka project ini LCD
 - Masukkan program ke flash Rom
 - Lepas jumper mode (run / program)
5. Untuk modul DST-R8C II bukalah project praktek_serial_B
 - **File => open workspace => praktek_serial_B.hws**
 - Build **dengan** menekan **F7** atau masuk menu **build => build**
 - **Build All** jika pertama kali buka project ini LCD
 - Masukkan program ke flash Rom
 - Lepas jumper mode (run / program)

8. Real Time Clock (RTC)

Real Time Clock (RTC) yang digunakan adalah RTC serial DS1307. DS1307 adalah IC serial RTC dimana alamat dan data ditransmisikan secara serial melalui sebuah jalur data dua arah I2C. Karena menggunakan Jalur data I2C maka hanya memerlukan dua buah pin saja untuk komunikasi. Yaitu pin untuk data dan pin untuk sinyal clock. Sistem jalur data I2C adalah suatu standar protokol sistem komunikasi data serial yang dikembangkan oleh Philips dan cukup populer karena penggunaannya cukup mudah.

Pada dasarnya, pada sistem I2C terbagi atas dua bagian, yaitu suatu device yang bertindak sebagai pengontrol atau *Master* dan suatu device yang dikontrol atau *Slave*. Master dan Slave saling berkomunikasi melalui jalur data bus I2C.

Register DS1307

Add	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Fungsi	Range
00	CH	Puluhan detik			Satuan detik				Detik	00 – 59
01	0	Puluhan menit			Satuan menit				Menit	00 - 59
02	0	12 24	10 jam AM/PM	10 jam	Satuan jam				Jam	1 –12 00 - 23
03	0	0	0	0	0	Hari			Hari	01 - 07
04	0	0	10 tanggal		Satuan tanggal				Tanggal	01 - 31
05	0	0	0	10 bln	Satuan bulan				Bulan	01 - 12
06	Puluhan tahun			Satuan Tahun					Tahun	00 - 99
07	Out	0	0	SQWE	0	0	RS1	RS0	Kontrol	----
08 – 3F									RAM	00 - FF

CH : Clock Halt (1 halt)
SQWE : Square Wave Enabel
RS0/1 : Rate Select

Praktikum 8 Real Time Clock

Tujuan

- Membuat Jam digital dengan menggunakan modul DST-R8C v.3

Peralatan

- Modul DST-R8C v.3
- LCD M1632

Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Pasang LCD ke modul DST-R8C sesuai dengan jenis LCD yang digunakan
3. Bukalah project praktek_rtc
File => open workspace => praktek_rtc.hws
4. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
5. Masukkan program ke flash Rom dan jalankan.

9. Serial EEPROM

Sebuah IC memori dengan kapasitas 2 KiloByte yang dibentuk dengan teknik transfer data secara paralel paling tidak mempunyai 24 kaki, 8 kaki untuk jalur data, 11 kaki untuk jalur penomoran memori (jalur alamat), 3 kaki untuk jalur kontrol, 2 kaki untuk catu daya.

Memori yang sama kalau dibentuk dengan teknik transfer data secara seri mempunyai 8 kaki, dan hanya 2 atau 3 kaki yang perlu dihubungkan ke Mikrokontroler. Dengan demikian IC dengan teknik transfer data secara seri banyak dipakai dalam rancang bangun peralatan berbasis Mikrokontroler.

Karena IC I2C hanya dikendalikan lewat kaki **SDA** dan **SCK** saja, tidak ada sarana lainnya dari microcontroller yang bisa dipakai untuk mengendalikan I2C, maka alamat yang dipakai untuk memilih isi IC I2C dikirimkan secara serial pula, persis seperti halnya pengiriman data pada serial RTC.

Pengalamat dasar IC I2C dilakukan dengan Nomor Group dan Nomor Chip.

- Nomor Group adalah nomor kelompok IC I2C. Sebagai contoh nomor group untuk Serial EEPROM adalah 1010 (biner).
- Nomor Chip adalah nomor yang diberikan pada masing-masing chip lewat kaki **A0**, **A1** dan **A2** dari masing-masing IC. Dalam IC I2C tertentu, **A0..A2** tidak dihubungkan ke kaki IC, tapi dipakai didalam IC untuk menomori register/memori di dalam IC bersangkutan

Praktikum 9 Serial EEPROM

Tujuan

- Menulis data ke dalam EEPROM dan membaca data dari EEPROM

Peralatan

- Modul DST-R8C v.3
- LCD M1632

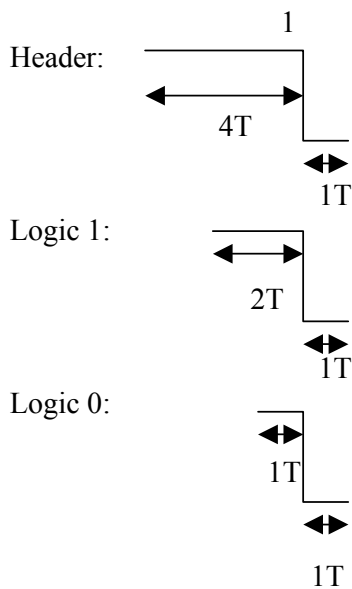
Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Pasang LCD ke modul DST-R8C sesuai dengan jenis LCD yang digunakan
3. Bukalah project praktek_rtc
File => open workspace => praktek_see.hws
4. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
5. Masukkan program ke flash Rom dan jalankan.

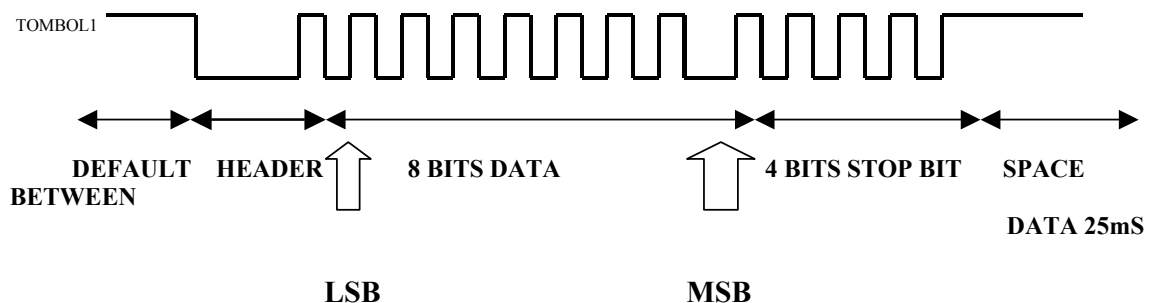
10. Remote Kontrol

Remote Sony ini memiliki karakteristik yaitu memiliki periode ($1T$)= $550 \mu s$ dan carrier $40KHz$. Untuk remote Sony memiliki header high $4T$ dan low $1T$, untuk logic 1 memiliki pulsa high sepanjang $2T$ dan low $1T$, dan untuk logic 0 memiliki pulsa high $1T$ dan low $1T$. Ini merupakan format aslinya sedangkan jika mengamati sinyal yang dikirimkan remote melalui IR modul kebalikannya karena pada IR modul ada inverternya.

Berikut contoh bentuk gambar pulsa dari header, logic 1 dan logic 0 dari remote TV Sony yang sebenarnya (belum diinvert) :



Untuk dapat mengamati bentuk sinyal yang dipancarkan oleh remote maka diperlukan osiloskop. Dengan osiloskop akan diketahui bentuk sinyal dari masing-masing tombol pada remote.



* Gambar setelah di invert

Praktikum 10 Remote Kotrol

Tujuan

- Menampilkan data yang dikirim oleh remote di LCD

Peralatan

- Modul DST-R8C v.3
- LCD M1632
- IR 8510
- Remote TV Sony

Prosedure Praktikum

1. Instalasi modul, pastikan modul pada mode program.
2. Pasang LCD ke modul DST-R8C sesuai dengan jenis LCD yang digunakan
3. Hubungkan IR 8510 dan DST-R8C sesuai tabel berikut

IR8510	DST-R8C
Pin 1	Ground
Pin 2	P1_0
Pin 3	Vcc (5 volt)

4. Bukalah project praktek_remote
File => open workspace => praktek_remote.hws
5. Build dengan menekan **F7** atau masuk menu **build => build**
Build All jika pertama kali buka project ini
6. Masukkan program ke flash Rom dan jalankan.
7. Tekan tombol remote

rev agustus 2007